**Team Based Development – DWCA Approach**



**Prepared By**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Andrew Siriwatt |  |
|  |  |
|  |  |

**Project Closure Report Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 0.1 | 8/22/2014 | AS | Initial |
| 1.0 | 11/06/2014 | AS | Included check in/check out screenshots |
| 1.1 | 1/06/2015 | AS | Added sections 5 and 6 |
| 1.2 | 1/07/2015 |  | Additional changes |

**TABLE OF CONTENTS**

[1 Purpose of this document 3](#_Toc408399280)

[2 Current Process 3](#_Toc408399281)

[2.1 Roles/Responsibilities 3](#_Toc408399282)

[2.2 Previous Activities Required for UAT/Production Migration 3](#_Toc408399283)

[2.3 New Activities Required for UAT/Production Migration: 4](#_Toc408399284)

[2.4 Process Flow 5](#_Toc408399285)

[3 Check In/ Check Out Process 6](#_Toc408399286)

[3.1 High Level Process to Develop 6](#_Toc408399287)

[3.2 Developer Instructions on How to Check Out/Check In 6](#_Toc408399288)

[3.3 Running in a Test Repository 8](#_Toc408399289)

[4 Migration 9](#_Toc408399290)

[4.1 Adding to deployment groups 9](#_Toc408399291)

[5 Stress Test 11](#_Toc408399292)

[5.1 Test 1: Migration of 2189 objects (38 mins): 11](#_Toc408399293)

[5.2 Stress Test #2 – 8000 objects 18](#_Toc408399294)

[6 Issues/Troubleshooting: 19](#_Toc408399295)

[6.1 Version control 19](#_Toc408399296)

[6.2 Migration Issues 20](#_Toc408399297)

# Purpose of this document

This document is to present team based development to improve our coding process and migration during development and testing.

# Current Process

## Roles/Responsibilities

Below is a high level description of the roles and the responsibilities each person has during lifecycle development. One person can have multiple roles except the Lead Migrator.

|  |  |
| --- | --- |
| **ROLE** | **RESPONSIBILITY** |
| Developer | The developer will have the ability to make changes to the development and testing regions. There is a checkout process that is integrated with the platform so any code change will require a check out.  When the developer is ready, they will request a deployment to either a Migrator or Lead Developer. |
| Lead Developer | The lead developer has developer rights but also oversees the work of the developer. They will be able to create the deploy groups for the developers.  Leads will be in charge of maintaining coding standards and managing cross impacts. |
| Migrator (to non UAT and Production environments) | This role is to create deployment groups based on the request of a developer. This role is to consolidate multiple requests and logically group them into deployment groups if possible.  Migrators will also be responsible for keeping track of deployment groups so they will be ready for UAT. |
| Lead Migrator | This role migrates the code from test into UAT and production. This role only has write access to these environments and not the development and testing regions. This role cannot have any other roles attached to it. |

## Previous Activities Required for UAT/Production Migration

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Developer** | **Lead** | **Migrator** |
| Check-out/check-in objects to VSS | X |  |  |
| Labeling objects in VSS | X |  |  |
| Identify cross impacts | X | X |  |
| Migration installation plan creation | X | X |  |
| UAT migration validation | X | X |  |
| UAT data validation | X |  |  |
| Production migration validation | X | X |  |
| Object count validation of VSS objects |  | X | X |
| Labeling validation for VSS objects |  | X | X |
| Importing code from VSS |  |  | X |
| Packaging scripts for migration |  |  | X |
| Migration to UAT |  |  | X |
| Remigrations to UAT |  |  | X |
| Installation plan additions for remigrations | X | X |  |
| Package for production |  |  | X |
| Migration to production |  |  | X |

## New Activities Required for UAT/Production Migration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activity** | **Dev** | **L Dev** | **Mig** | **L Mig** |
| Check out object – development region | X | X |  |  |
| Check in object – development region | X | X |  |  |
| Development | X | X |  |  |
| Create deploy group |  | X | X |  |
| Add to deploy group | X | X |  |  |
| Migration to test region |  |  | X |  |
| Check out object – test region |  | X | X |  |
| Check in object – test region |  | X | X |  |
| Installation plan for new migration | X | X |  |  |
| Migration to UAT environment |  |  |  | X |
| Migration to production environment |  |  |  | X |
| Manage deploy groups – UAT/PROD |  | X | X |  |

## Process Flow



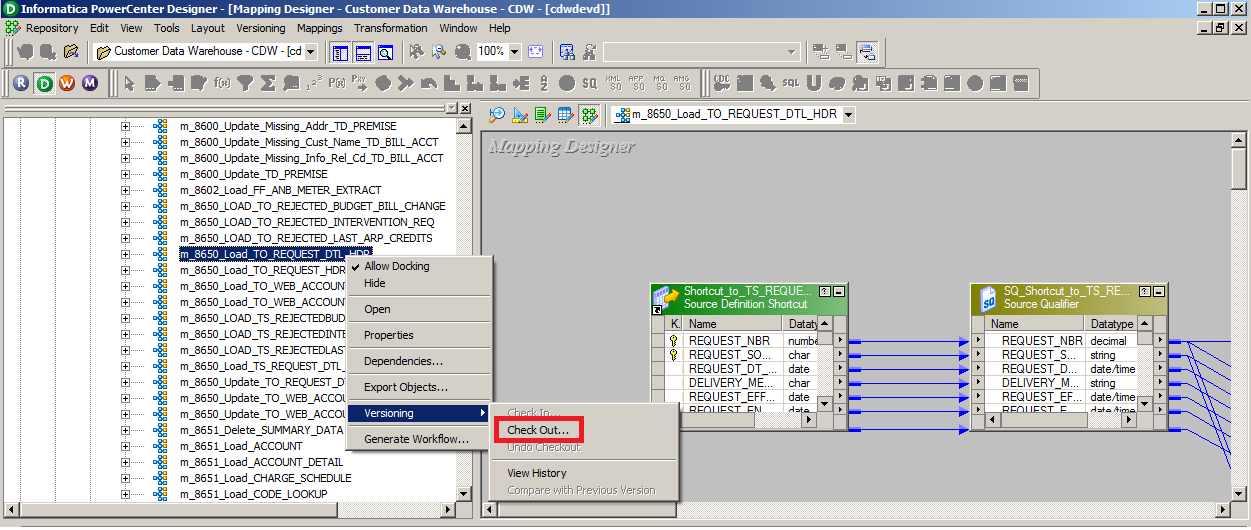
# Check In/ Check Out Process

## High Level Process to Develop

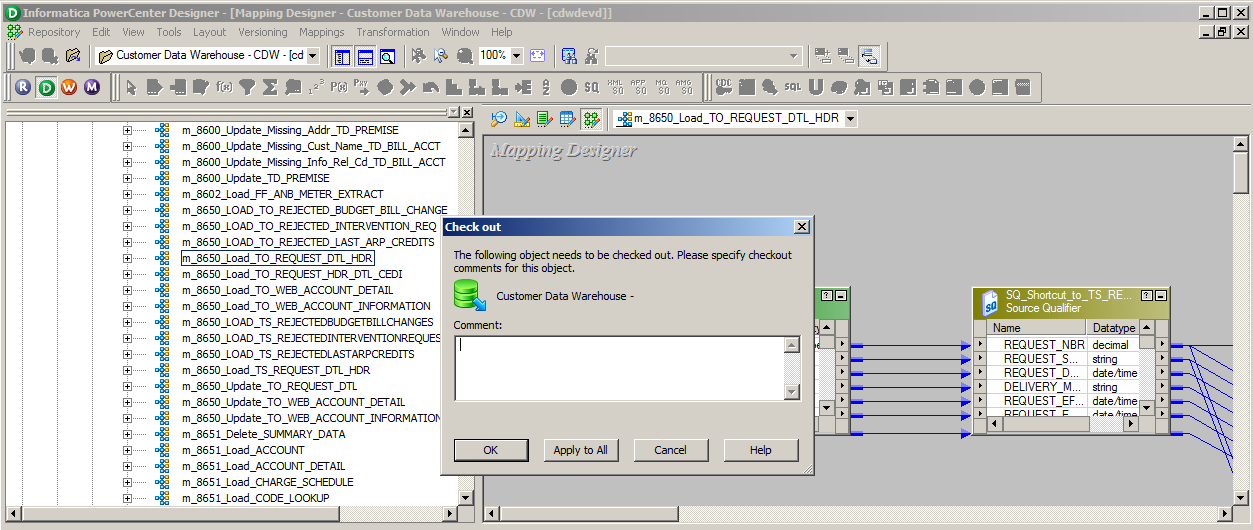
|  |  |
| --- | --- |
| Projects | Maintenance |
| 1) Start check out in CDWDEVA | 1) Start check out in CDWDEVA |
| 2) Development in CDWDEVA project folders | 2) Development in CDWDEVA project folders |
| 3) Move code to CDWDEVB for test conditions | 3) Move code to CDWDEVC for test conditions |
| 4) Do not code in CDWDEVB – only for connection changes (There will be a daily check for this) | 4) Do not code in CDWDEVC – only for connection changes (There will be a daily check for this) |
| 5) Any other region will follow the same steps above except CDWDEVB will be replaced with CDWDEV% | 5) Any other region will follow the same steps above except CDWDEVC will be replaced with CDWDEV% |
|  |  |

## Developer Instructions on How to Check Out/Check In

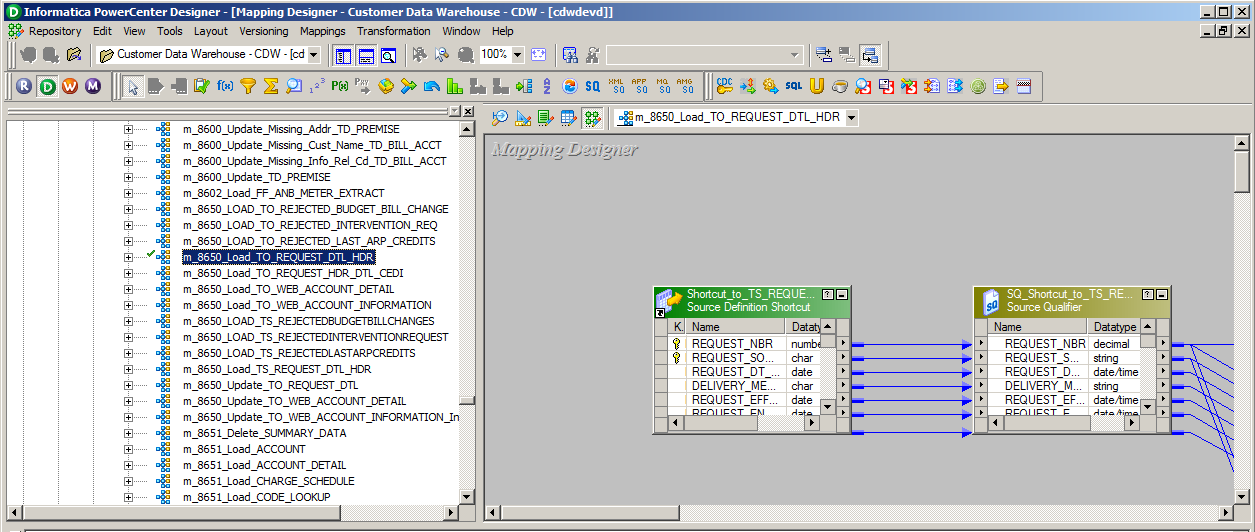
Right-click on object and go to Versioning->Check Out…



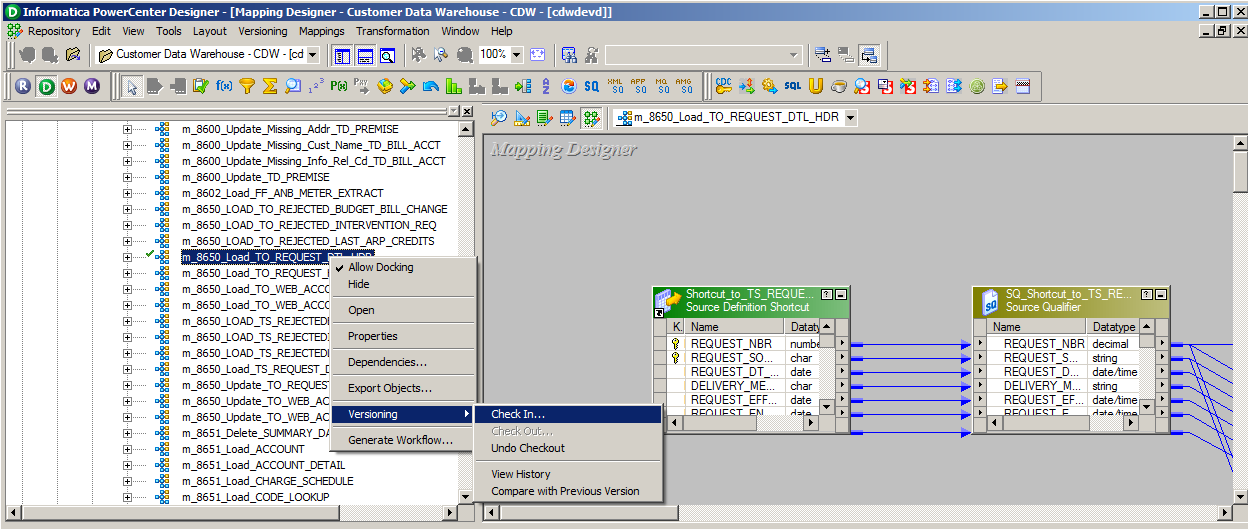
Entering a comment is not required at check out. However it is required at the check in, so you can either do it then or here. Please include the following format SIR##### - <description>



Check mark next to the object name will appear – this means that this object has been checked out.



To check in, right-click on object and go to Versioning->Check In…



Please include the following format SIR##### - <description>. If you did this previously during the Check Out, this information will already be populated.

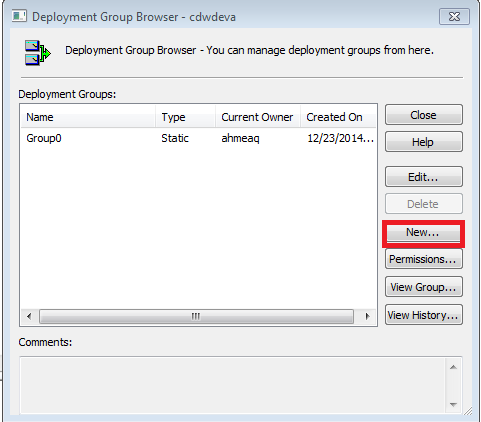
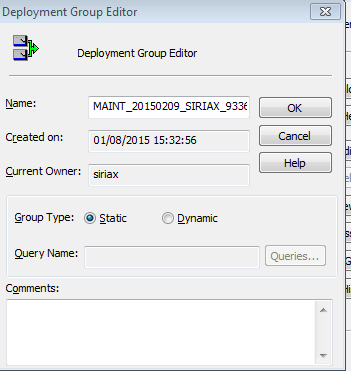
## Running in a Test Repository

When running in the test repository, developers will have the ability to modify connections. You will need to check out the workflow and/or the session. Remember to undo the checkout once you are done and the connections have been reverted.

# Creating a Deployment Group

## Creating deployment groups

Click on NEW

 🡪 

For Projects team: PROJ\_<PUSHDATE>\_<USERID>\_<SIR#>

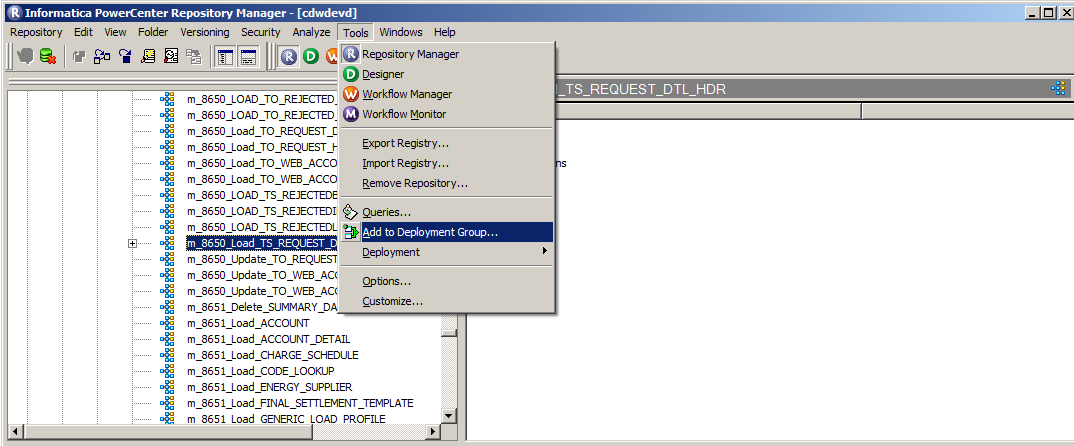
For Maintenance team: MAINT\_<PUSHDATE>\_<USERID>\_<SIR#>

Pushdate is in YYYYMMDD format

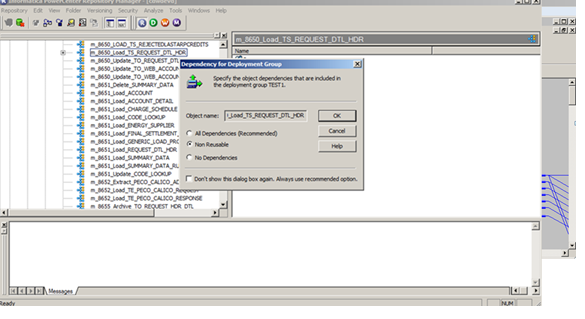


## Adding to deployment groups

In Repository Manager, right click on an object and select ‘Add to Deployment Group’



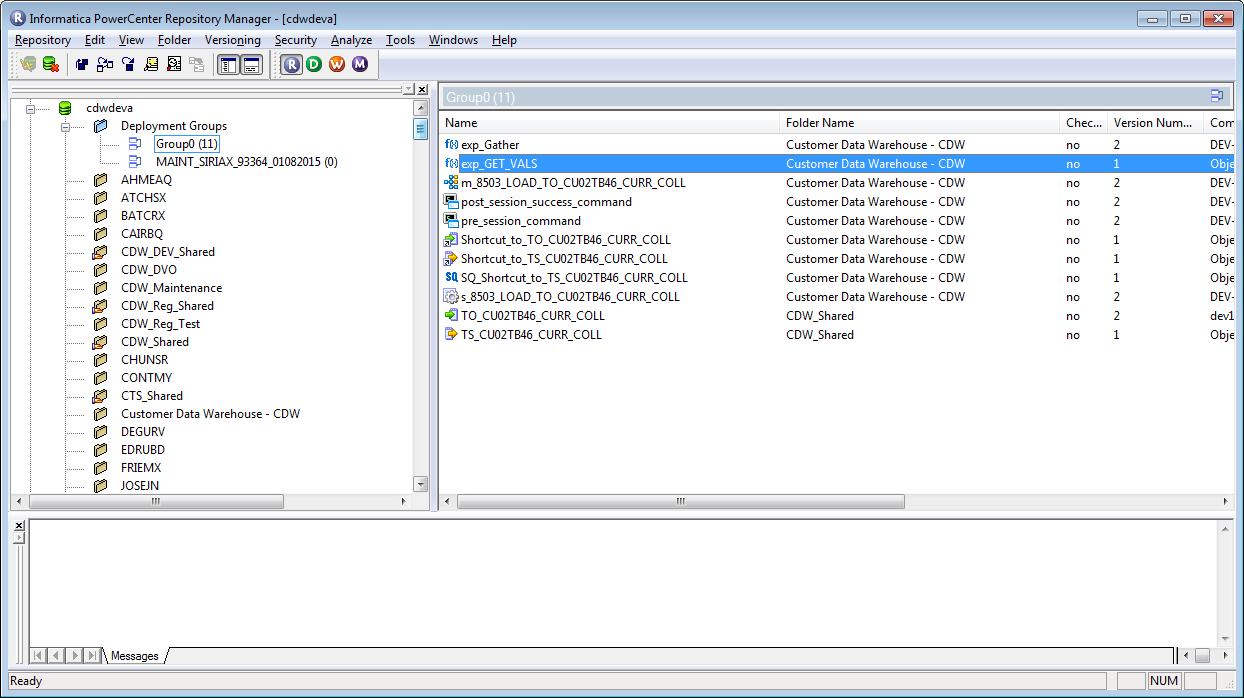
When moving from CDWDEVA to a testing repository, choose the following option – Non-Reusable



If for some reason there is a migration issue, we will redeploy with all dependencies.

## Removing from deployment group

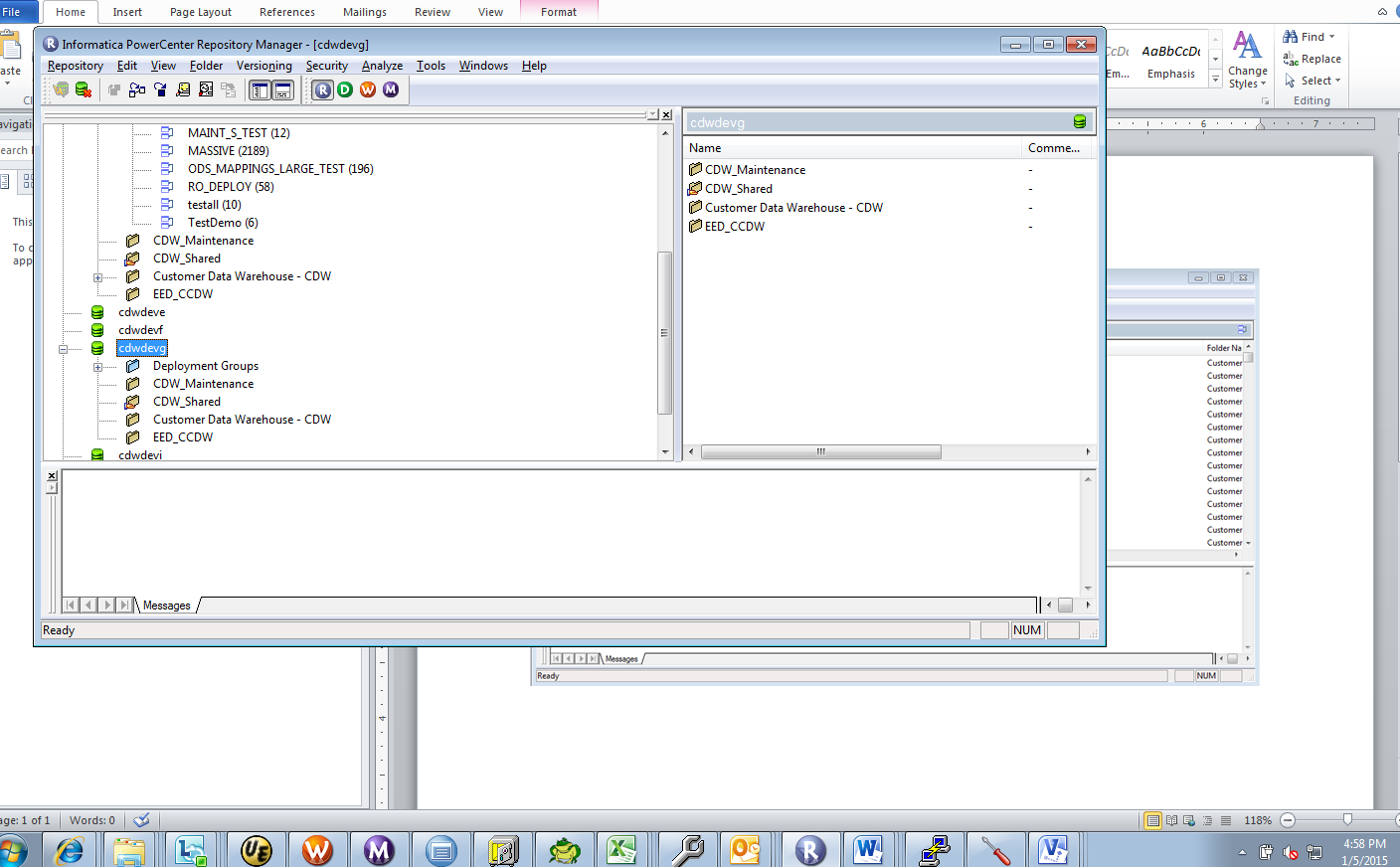
Open up deployment group, right click on the object and select ‘Remove’



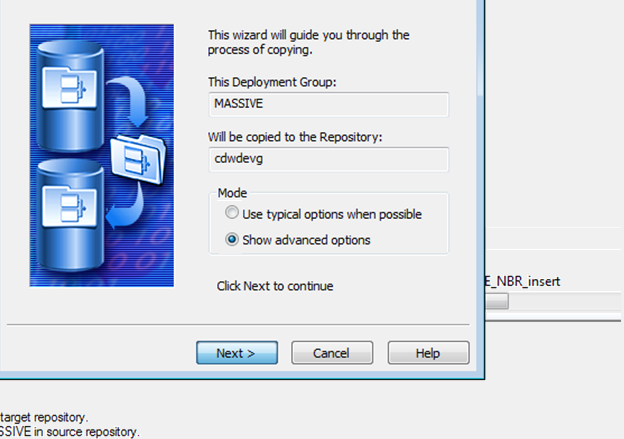
# Migration

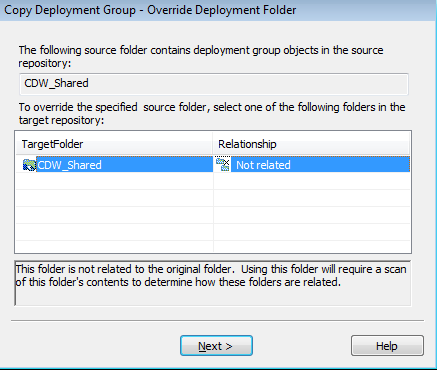
## Test 1: Migration of 2189 objects (38 mins):

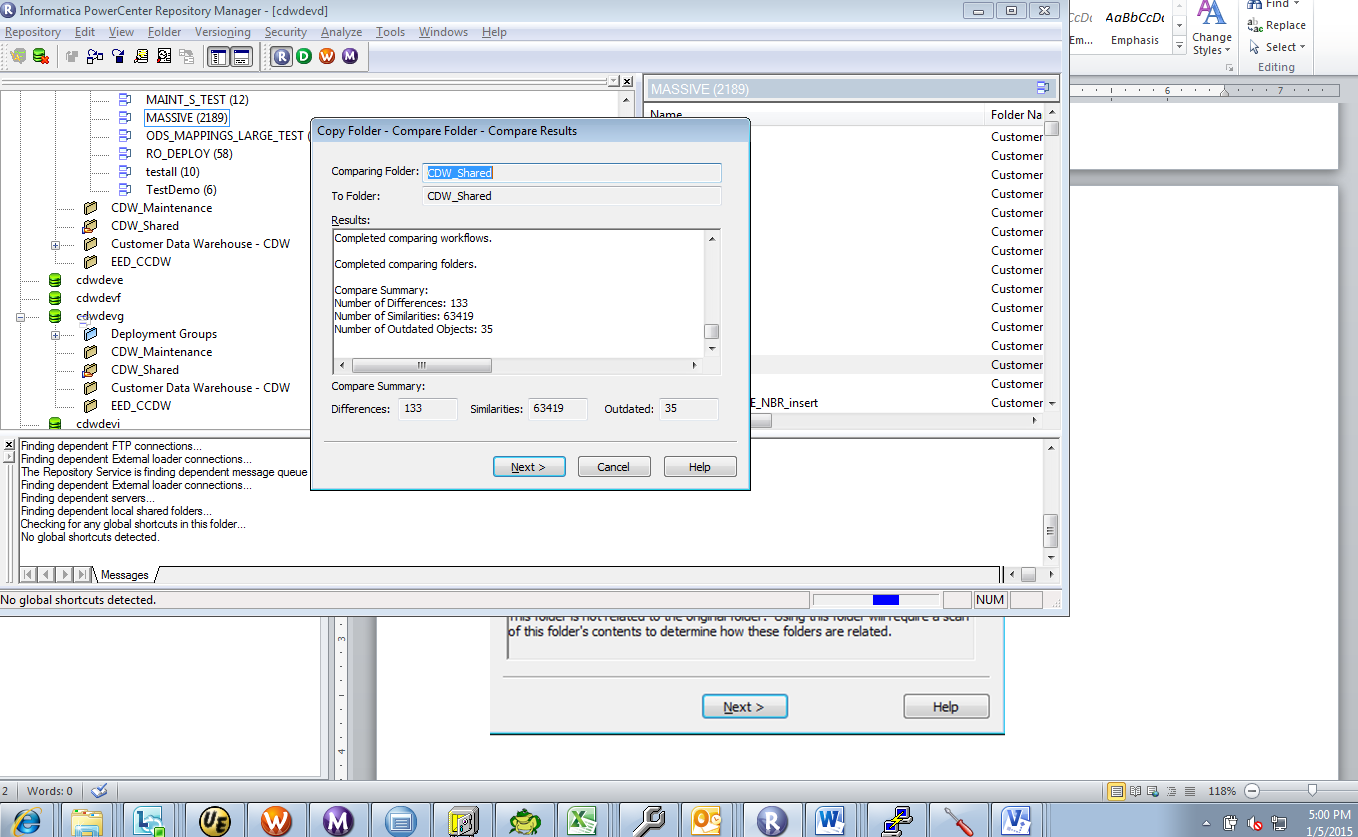
Time started 4:58 PM. Selecting a deployment group from one repository to another (CDWDEVD to CDWDEVG).

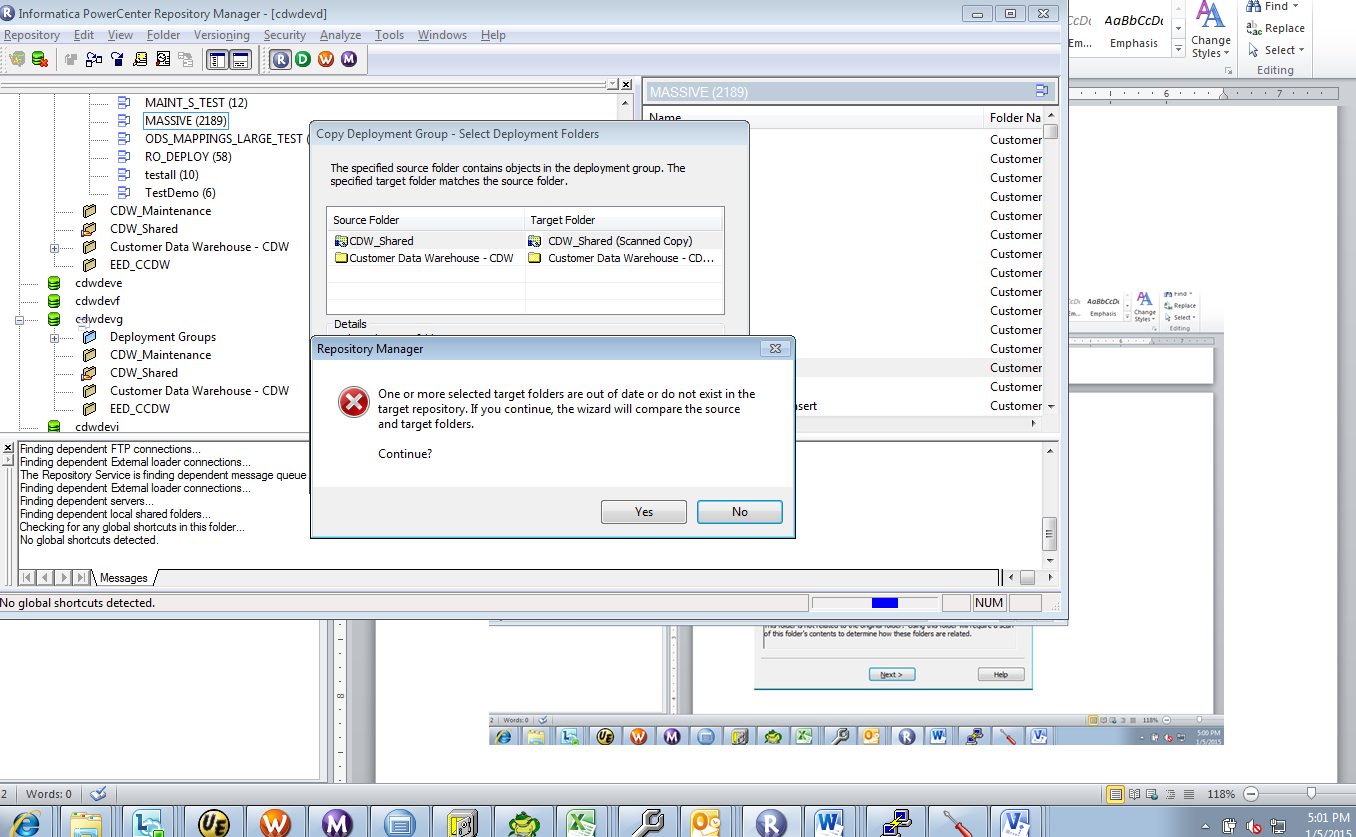


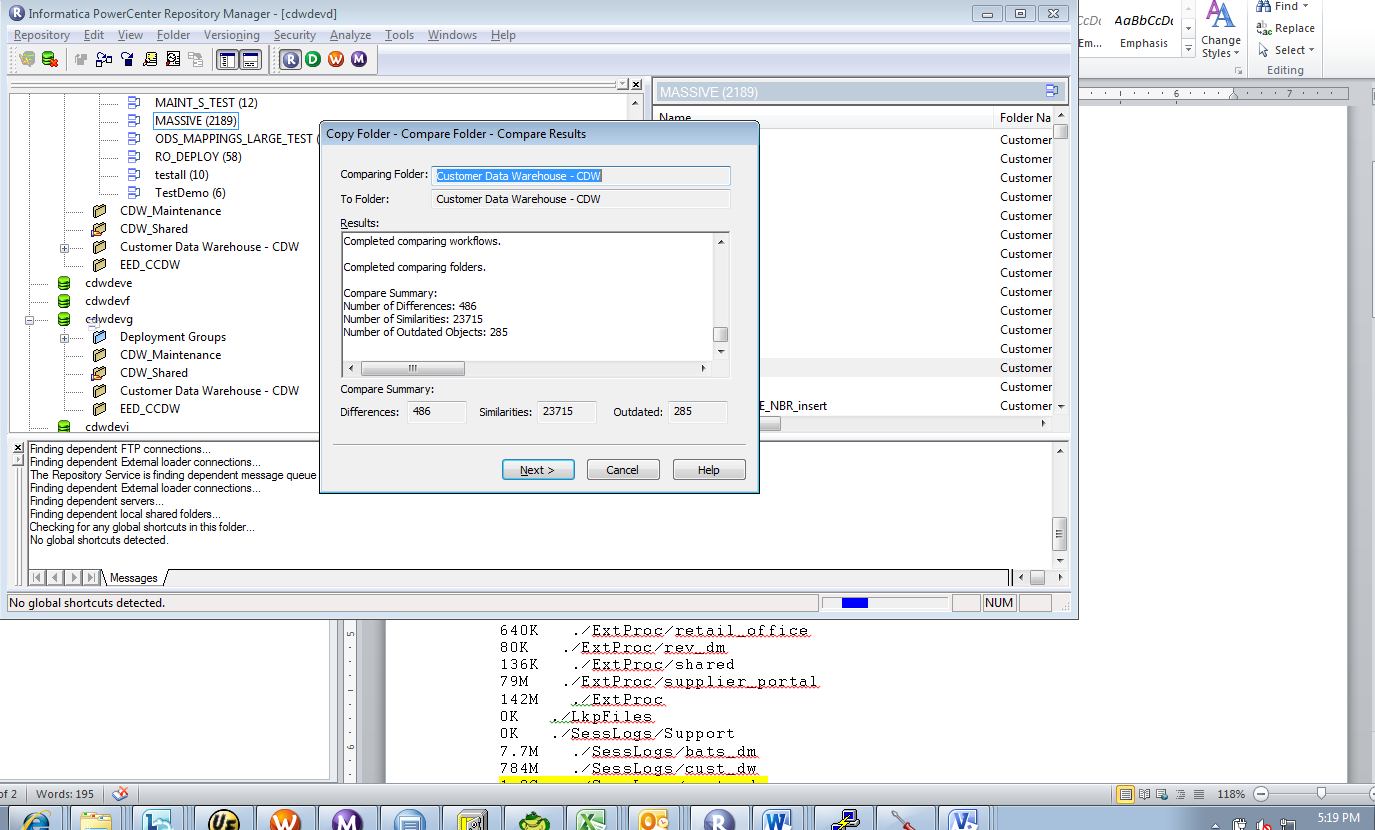
After dragging a deployment group to cdwdevg, a wizard pops up. The following screenshots are in successive order of the migration wizard.



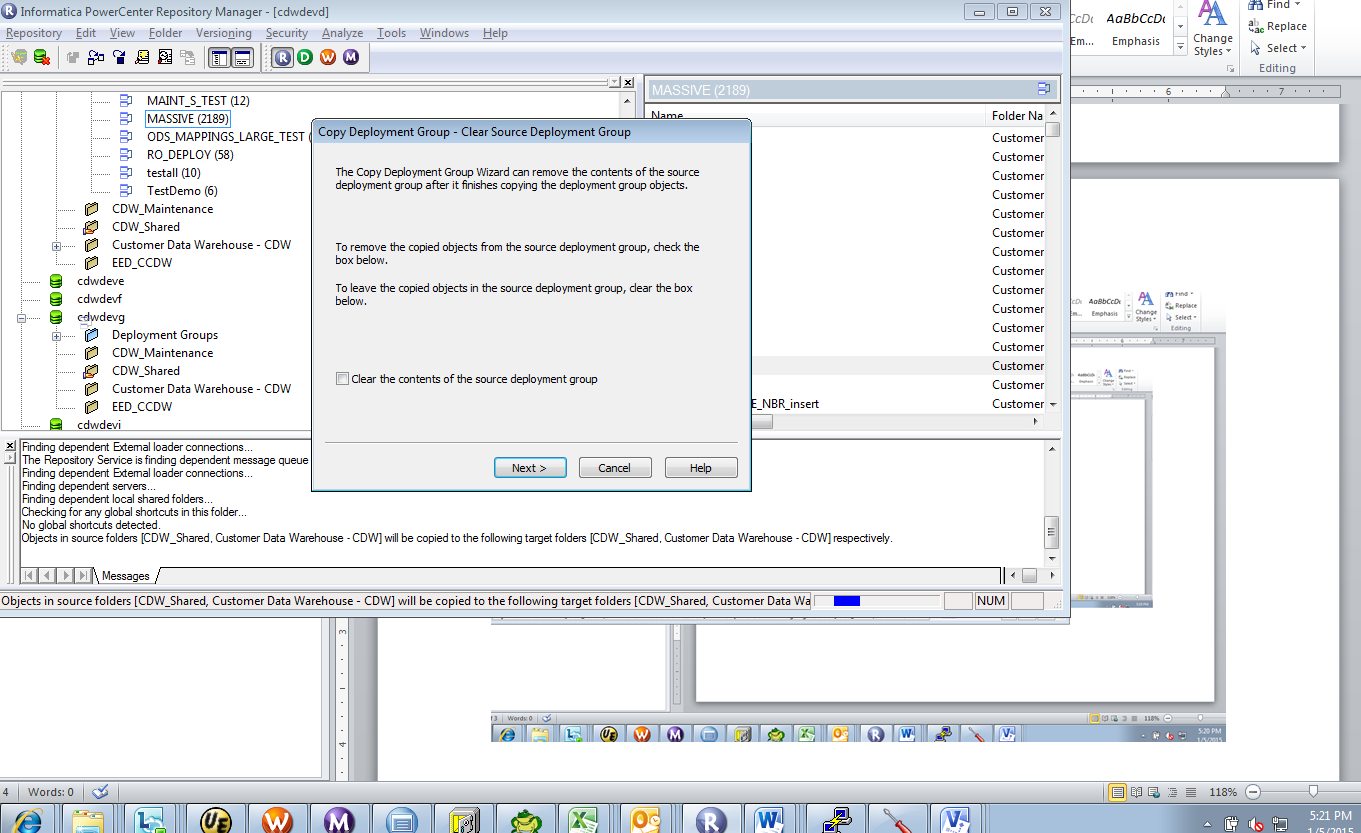




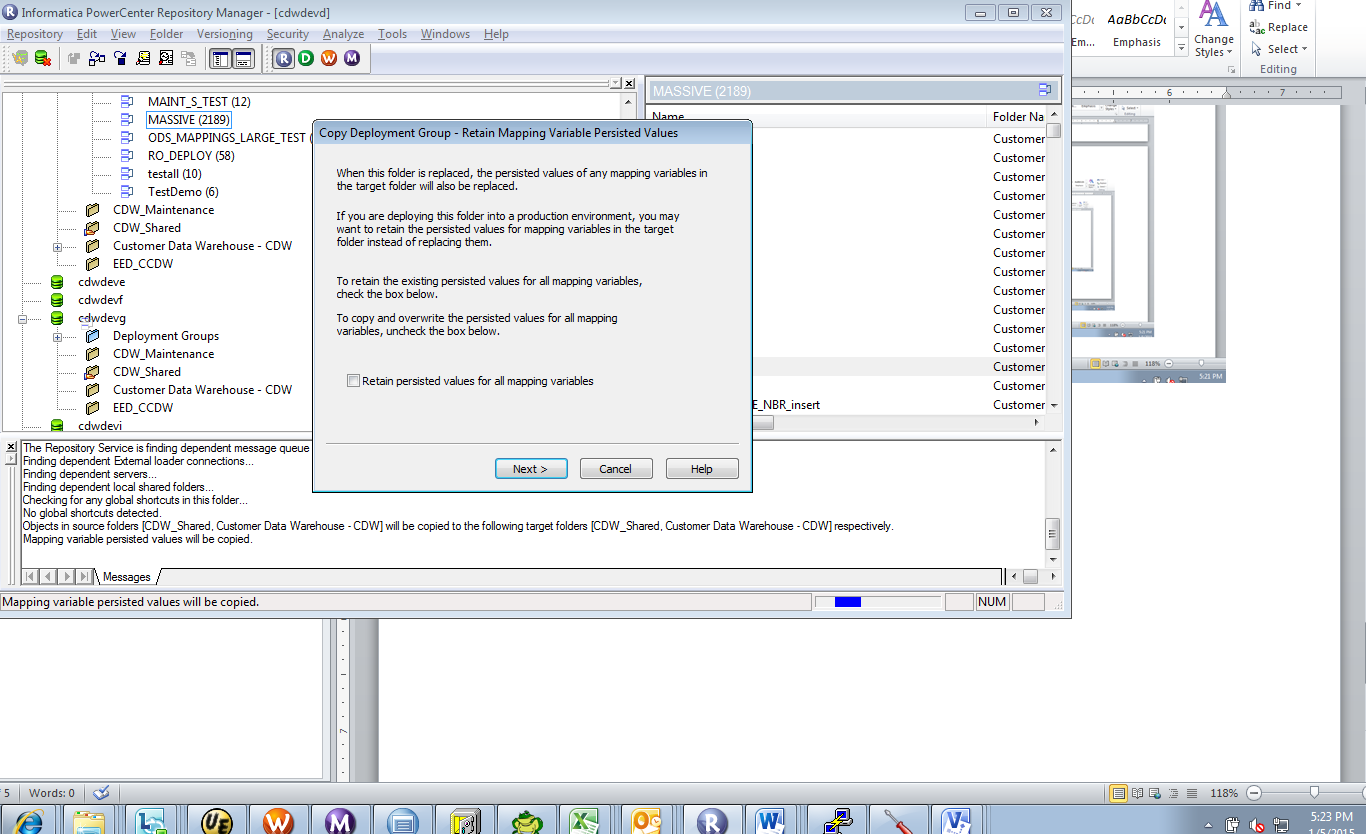


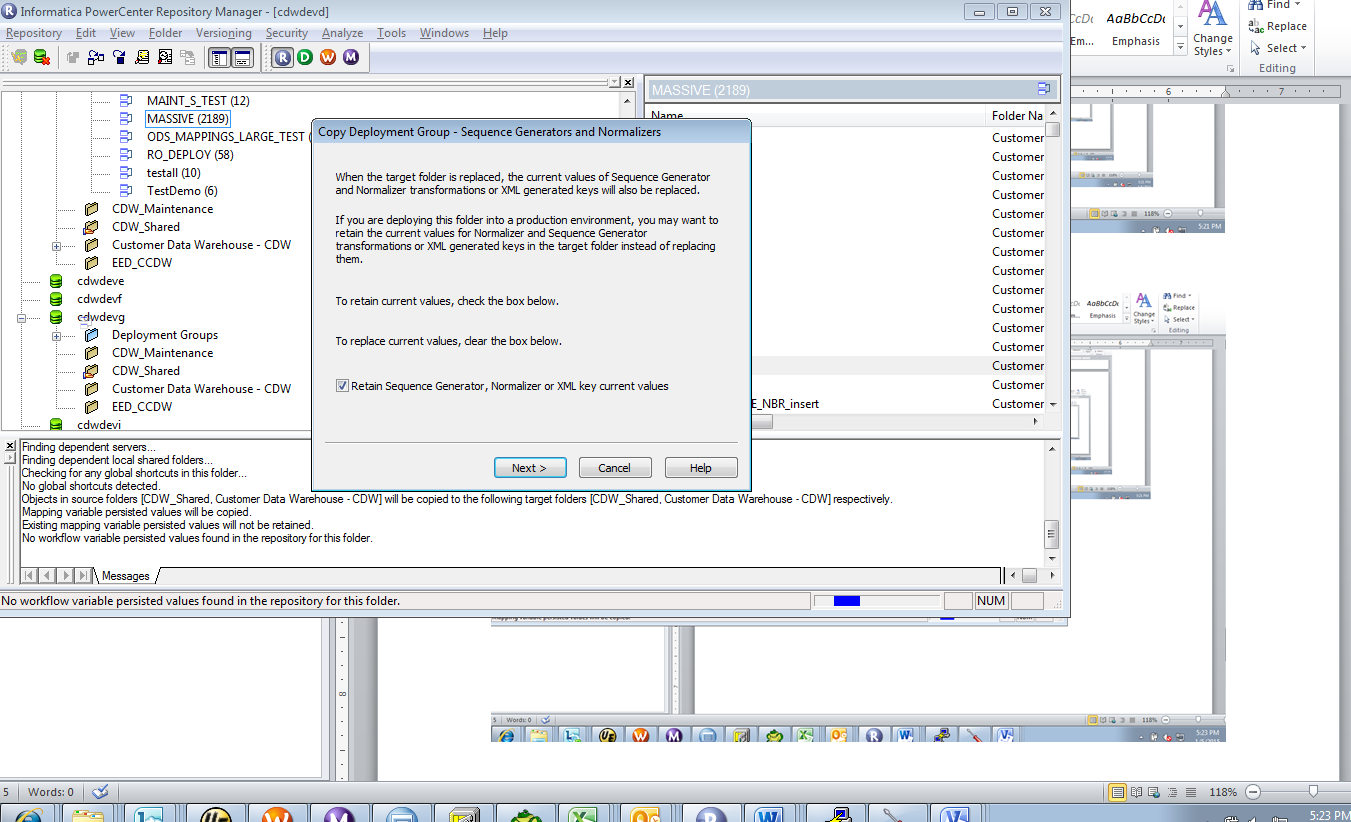


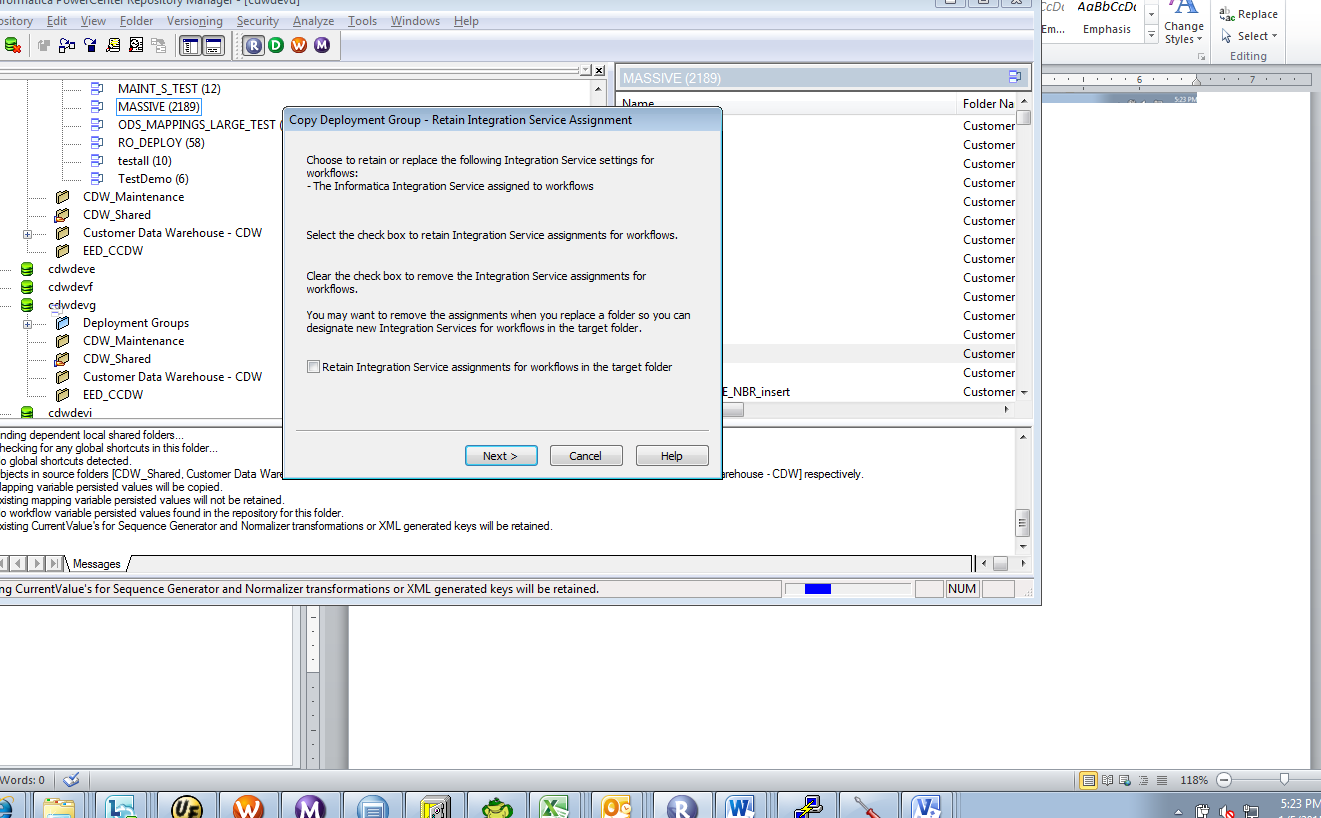


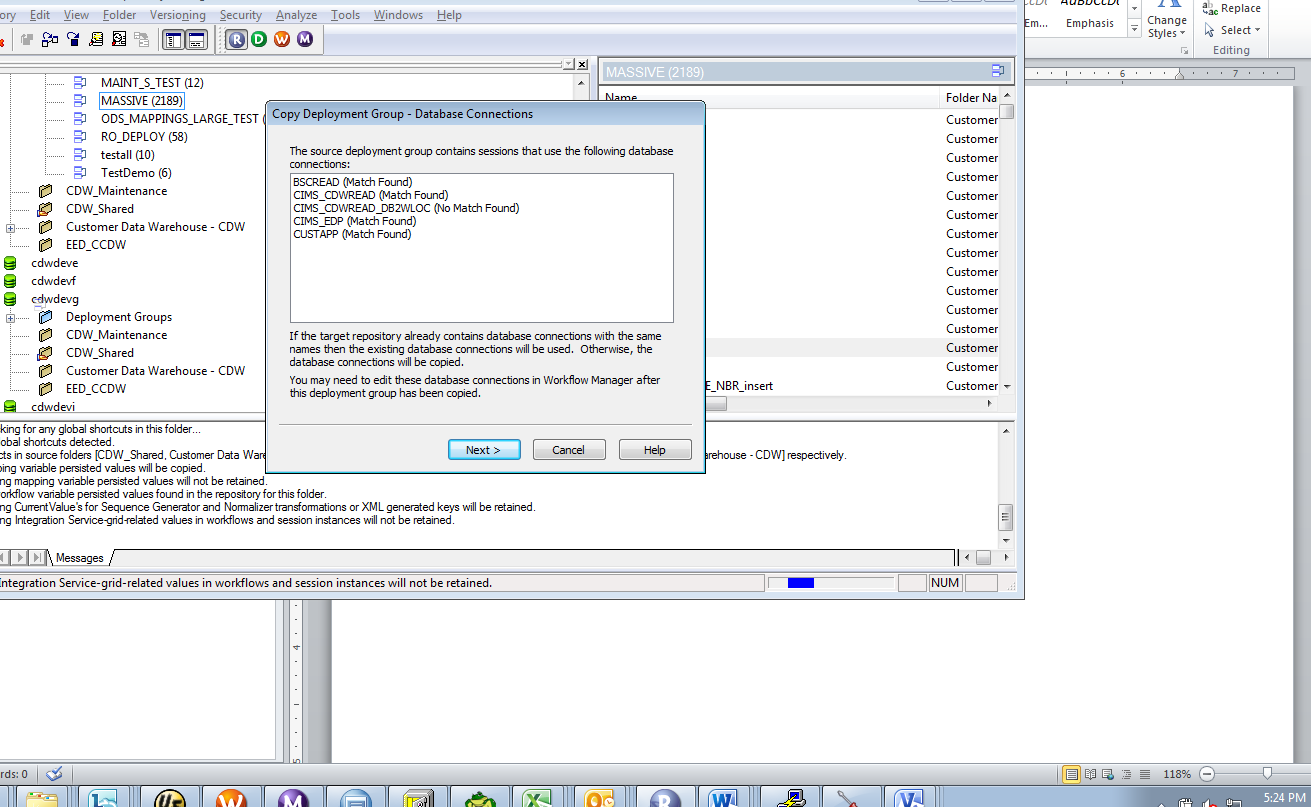




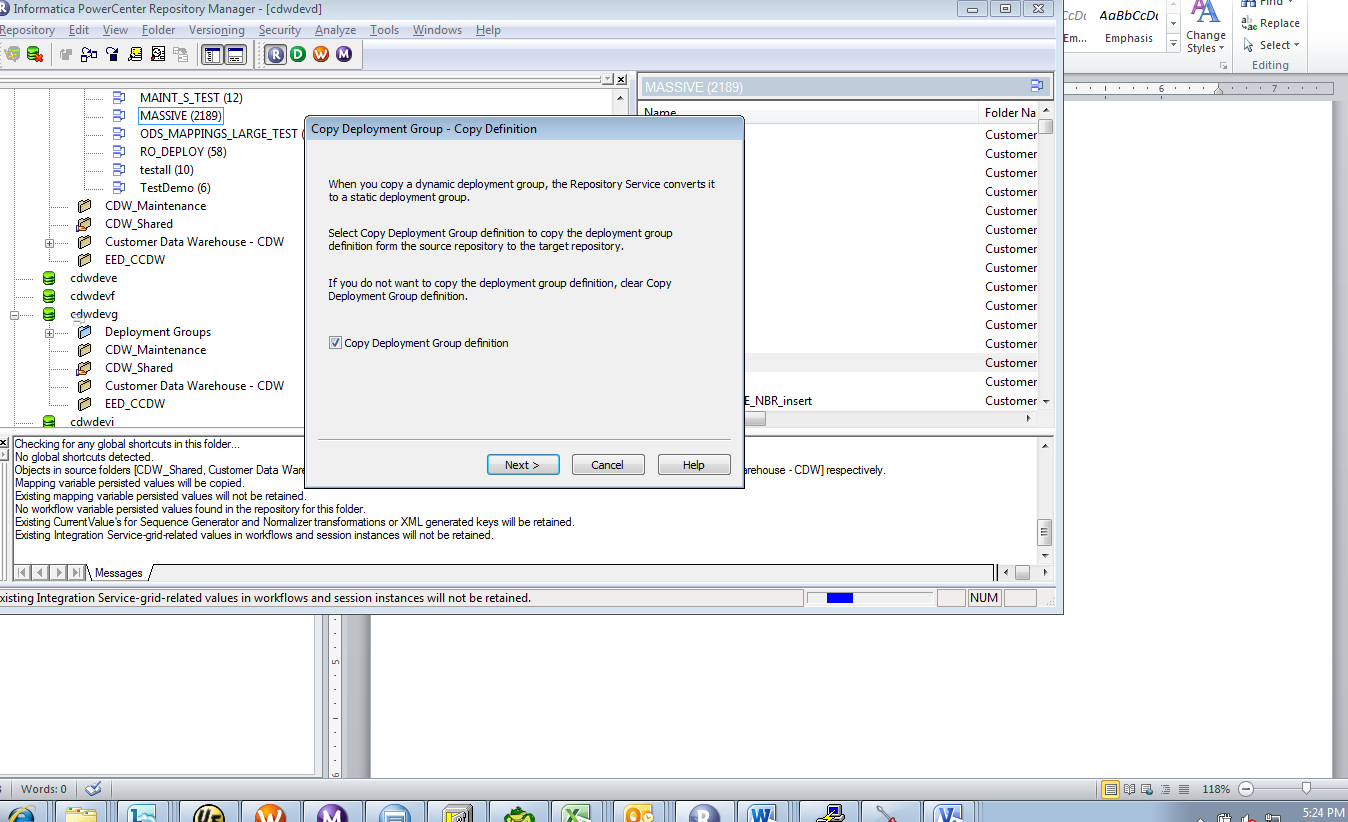


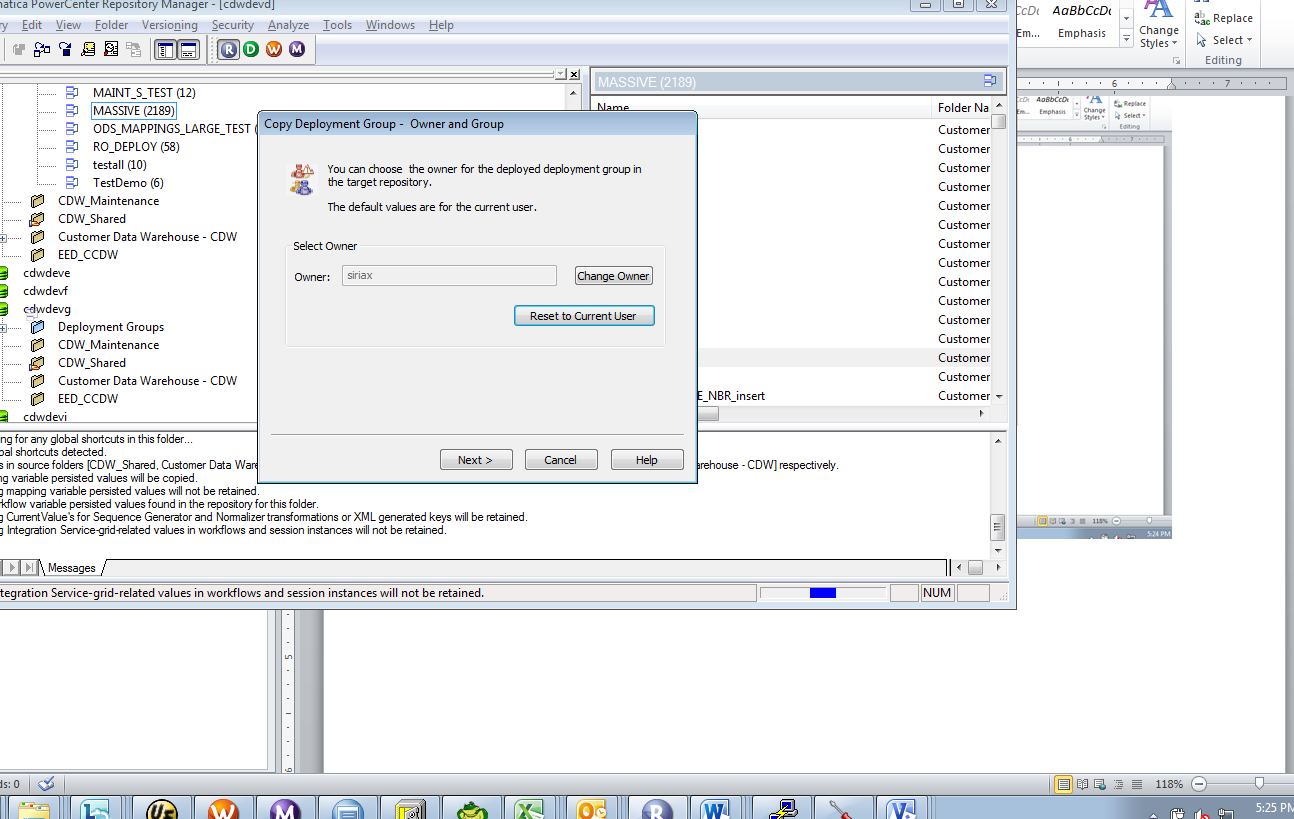




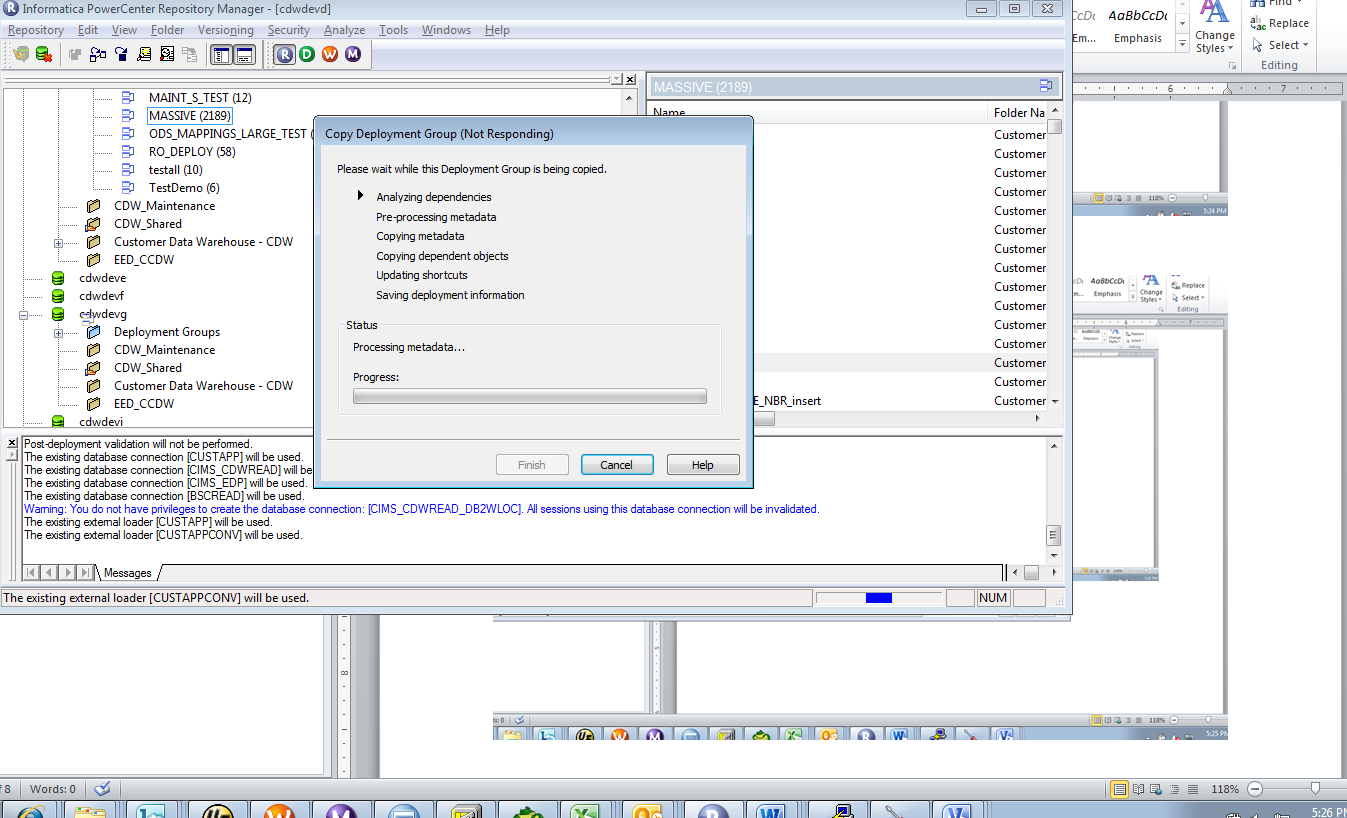






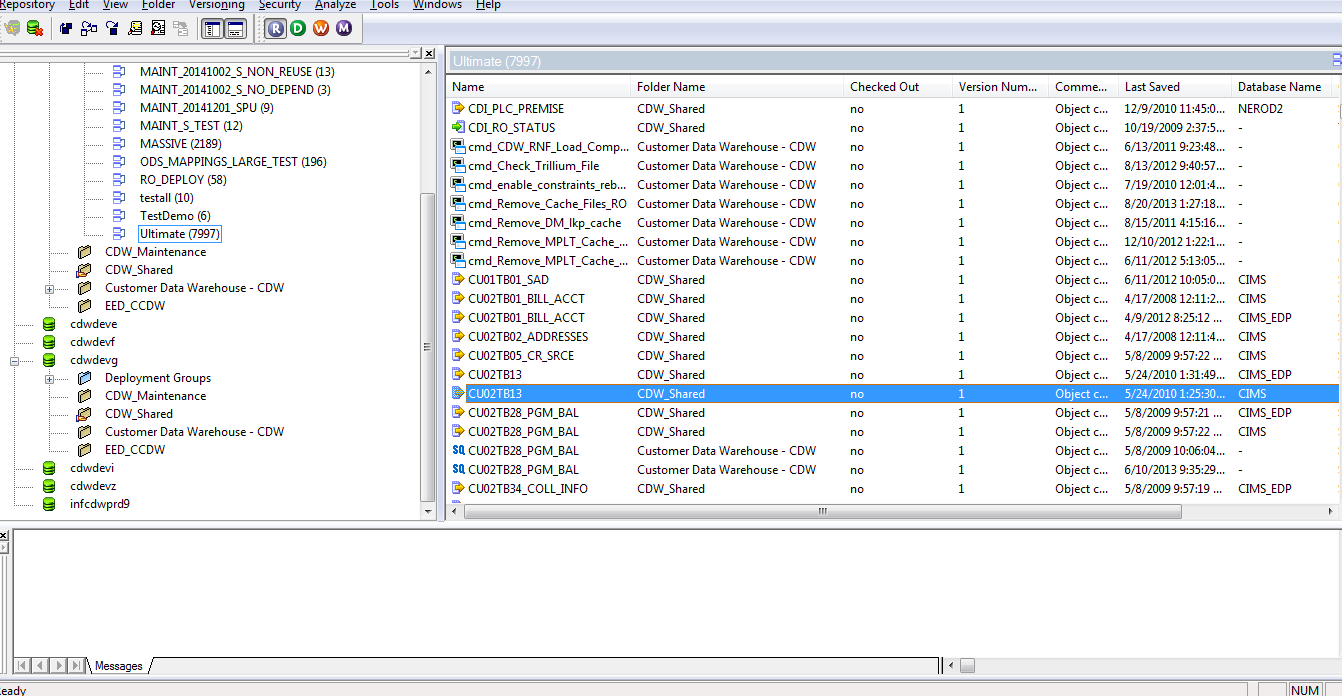


Note that I did not have connection creation privileges so I was not able to create a connection to the target repository (warning in blue). This won’t be a problem in UAT since all connections should be in all environments. In the case below, the connection was a testing connection only in one of the environments and not one used in UAT/Production.

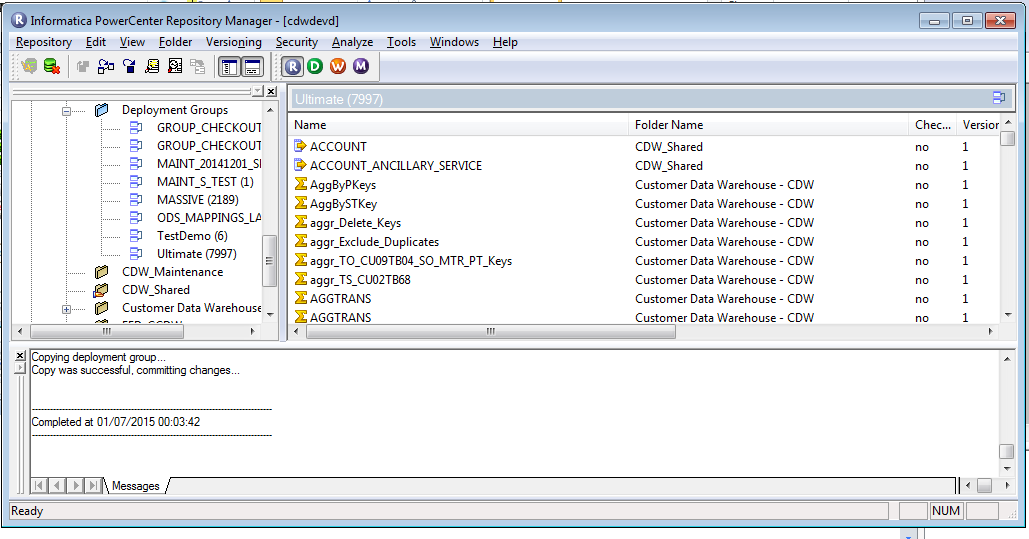




## Stress Test #2 – 8000 objects

This shows that the Ultimate group has 7997 objects (reusable and non resuable) in CDWDEVD.

Deploying to CDWDEVG was less than an hour.



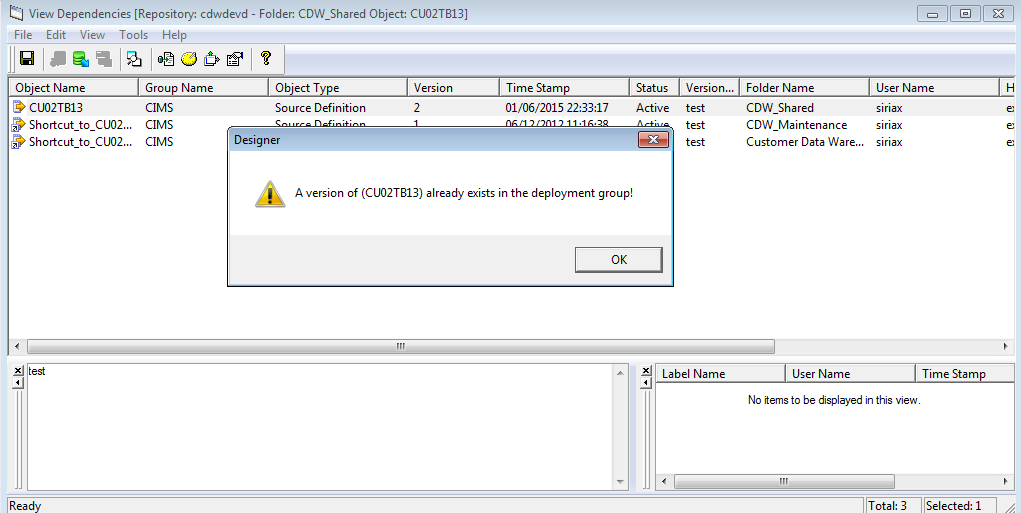
# Issues/Troubleshooting:

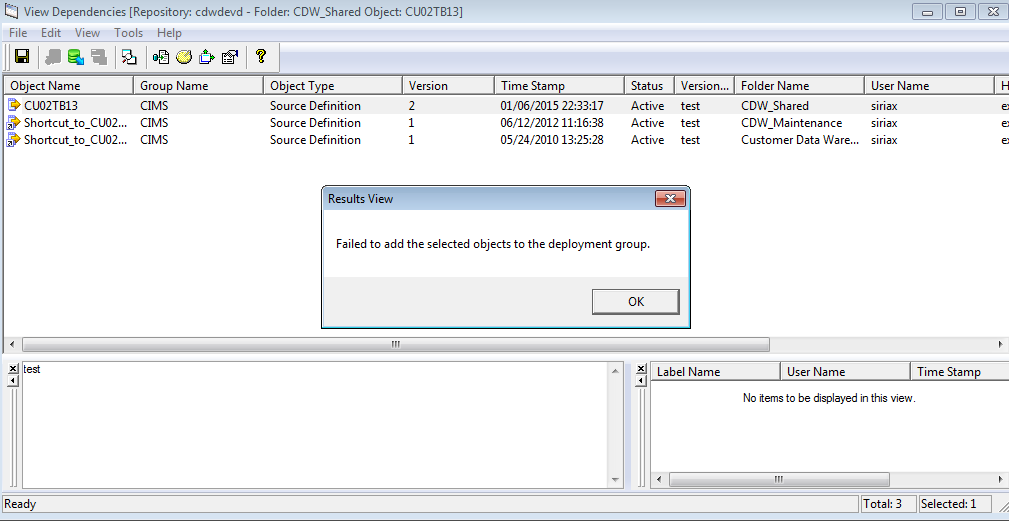
## Version control

Note if you add existing objects to a group you will get a warning mesage:



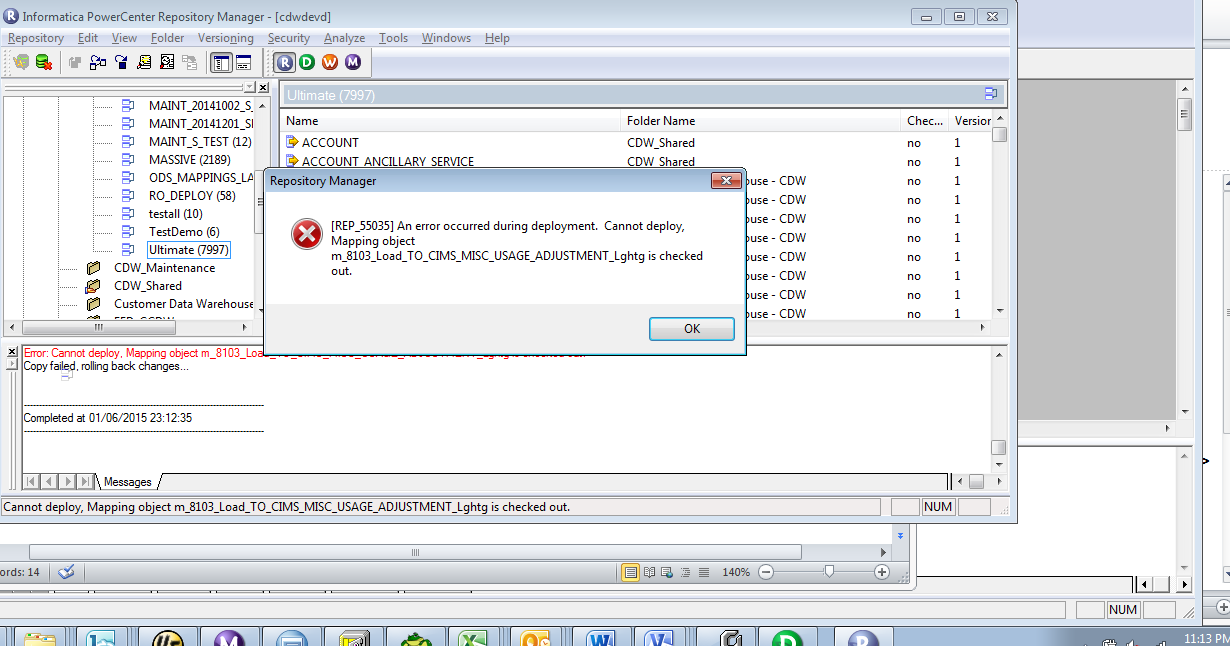
Also trying to add a new version of an existing object in that group is not allowed:





## Migration Issues

1 ) Make sure the object is not checked out in the target destination. If it is, migration will not proceed.

\*\*\*MIGRATORS should verify that the objects are not checked out in the target repository.\*\*\*

2) Make sure that nobody has an active session that is writeable in the target repository